WHAT IS CLAIMED IS:

1. A method of operating a computer system that includes first and second processors and memory shared thereby, the method comprising:

concurrently executing first and second instructions on respective ones of the first and second processors, the first and second instructions each reserving in a same predefined order plural respective locations of the memory,

wherein, for at least the first instruction, signaling of a fault corresponding to a later reserved one of the respective locations depends on a value read from an earlier reserved one of the respective locations.

2. The method of claim 1,

wherein the predefined order is in accordance with a fixed total order of locations within the memory.

3. The method of claim 1,

wherein the predefined order is one of ascending and descending memory address order.

4. The method of claim 1,

wherein the reserving includes locking an associated cache-line.

5. The method of claim 1,

wherein the reserving locks at least the respective location, but substantially less than all the memory.

6. The method of claim 1,

wherein the first and second instructions are linearizable synchronization operations.

7. The method of claim 1,

wherein the first instruction is a double compare-and-swap instruction.

8. The method of claim 1,

wherein the first instruction is a compound compare-and-swap instruction;

wherein the respective locations reserved by the compound compare-and-swap instruction number N; and

wherein signaling of a fault corresponding to a later reserved one of N locations depends on at least one value read from an earlier reserved one the N locations.

9. The method of claim 1,

wherein the first instruction is a double compare-and-swap instruction; and

wherein, unless a value read from the earlier reserved one of the respective locations compares to a corresponding test value, no fault corresponding to the later reserved one of the respective locations is signaled.

10. The method of claim 1,

wherein the execution of the first instruction includes access to the earlier reserved one of the respective locations; and

wherein, unless the access succeeds, no fault corresponding to the later reserved one of the respective locations is signaled.

11. The method of claim 1, wherein the execution of the first instruction includes

a first access corresponding to an earlier reserved one of the respective locations; and

an optional second access that signals a fault only if the first access succeeds.

12. The method of claim 1,

wherein the respective locations reserved by the first and second instructions are disjoint; and

wherein the concurrent execution is non-blocking.

13. A processor that implements an instruction that separately reserves plural memory locations in an order prescribed by a fixed total order of the memory locations.

14. The processor of claim 13,
wherein the fixed total order of the memory locations is one of ascending memory address order and descending memory address order.

15. The processor of claim 13,
wherein the separately reserved memory locations include an earlier reserved first memory location and a later reserved second memory location; and
wherein signaling of a fault corresponding to the second memory location depends on a value read from the first memory location.

16. The processor of claim 13,
wherein the instruction is a compound compare-and-swap instruction; and
wherein the instruction signals a fault corresponding to a later reserved one of the memory locations only if a value read from an earlier reserved one of the memory locations compares to a test value.

17. The processor of claim 13,
wherein the instruction signals a fault corresponding to a later reserved one of the memory locations only if access to an earlier reserved one of the memory locations succeeds.

18. The processor of claim 13, wherein access to at least the first memory location is one of:
a compare-and-swap access;
a test-and-set access;
a read-compute-conditional write access; and
a read-modify-write access.

19. The processor of claim 13,
wherein the reserving includes locking an associated cache line.

20. A computer system comprising:
first and second processors and storage shared thereby;
the first and second processors each implementing an instruction that separately reserves locations of the storage addressed thereby in an order prescribed by a fixed total order of the locations.

21. The computer system of claim 20,
wherein, on execution of the instruction, the locations addressed thereby include an earlier reserved first location and a later reserved second location; and
wherein signaling of a fault corresponding to the second location depends on a value read from the first location.

22. The computer system of claim 20,
wherein the instruction is a compound compare-and-swap operation; and
wherein the instruction signals a fault corresponding to a later reserved one of the locations only if a value read from an earlier reserved one of the locations compares to a test value.

23. The computer system of claim 20,
wherein instances of the instruction that address disjoint sets of the locations are concurrently executed by the first and second processors.

24. The computer system of claim 20, further comprising:
a coherently maintained set of caches including first and second caches respectively associated with the first and second processors,
wherein, if a first instance of the instruction and a second instance of the instruction address locations associated with distinct sets of cache lines, then the first and second instances of the instruction are concurrently executable by the first and second processors.

25. A computer program product encoded in at least one computer readable medium, the computer program product comprising:

a first set of instructions executable on a first processor,

the first set including at least one instance of a particular instruction directing the processor to separately reserve plural storage locations referenced thereby in a predefined order such that concurrent execution of the particular instruction and a corresponding instruction avoids deadlock, the corresponding instruction also separately reserving plural storage locations referenced thereby in the predefined order.

26. The computer program product of claim 25,

wherein the particular instruction itself provides reservation in the predefined order.

27. The computer program product of claim 25,

wherein the first set includes additional instructions to ensure that storage location references supplied to the particular instruction are ordered in accordance with the predefined order.

28. The computer program product of claim 25, further comprising:

a second set of instructions executable on a second processor,

the second set including at least one instance of the corresponding instruction.

29. The computer program product of claim 25,

wherein the predefined order is ascending memory address order.

30. The computer program product of claim 25,

wherein the predefined order is descending memory address order.

31. The computer program product of claim 25,

wherein the particular instruction and the corresponding instruction are same instructions.

32. The computer program product of claim 25,
wherein the particular instruction and the corresponding instruction each implement a compound Compare-and-swap (nCAS) operation.

33. The computer program product of claim 25,
wherein the at least one computer readable medium is selected from the set of a disk, tape or other magnetic, optical, or electronic storage medium and a network, wireline, wireless or other communications medium.

34. A computer system that allows two or more DCAS instructions to operate concurrently if they operate on distinct cache lines.

35. A processor that implements a first instruction that reserves, in an predefined order, plural storage locations referenced thereby, wherein concurrent execution of the first instruction and a corresponding instruction that also reserves, in the same predefined order, plural storage locations referenced thereby is non-blocking if the first and the corresponding instructions reference distinct memory portions.

36. The processor of claim 35, further implementing the corresponding instruction.

37. The processor of claim 35,
wherein the storage locations are sharable with a second processor; and
wherein the corresponding instruction is implemented at least by the second processor.

38. The processor of claim 35,
wherein the reserving includes locking the referenced storage locations.

39. The processor of claim 35,
wherein the reserving includes locking cache lines associated with the referenced storage locations.

40. A processor of claim 35,
wherein the predefined order is ascending memory address order.

41. A processor of claim 35,
wherein the predefined order is descending memory address order.

42. The processor of claim 35,
wherein the first instruction and the corresponding instruction are same instructions.

43. The processor of claim 35,
wherein the first instruction and the corresponding instruction both implement a compound Compare-and-swap (nCAS) operation.

44. An apparatus comprising:
a memory store; and
means for separately reserving in response to a single instruction, plural locations of the memory store in a predefined order in accordance with a fixed total order of locations in the memory store.

45. The apparatus of claim 44, further comprising:
a cache,
wherein the means for separately reserving includes means for separately locking cache lines associated with each of the plural locations.

46. The apparatus of claim 44, further comprising:
means for signaling, if at all, a fault corresponding to a later reserved one of the locations based on a result of access to an earlier reserved one of the locations.

47. A method of operating a computer system that includes a memory shared by plural processors thereof, the method comprising:
in response to execution of a single instruction by one of the processors,

separately reserving plural locations of the memory; and

signaling a fault corresponding to a later reserved one of the locations based on a value read from an earlier reserved one of the locations.

48. The method of claim 47,

wherein, unless the value read from the earlier reserved location compares to a corresponding test value, no fault corresponding to the later reserved location is signaled.

49. The method of claim 47,

wherein the separately reserving includes separately locking at least the plural locations, but substantially less than all the memory.

50. The method of claim 47,

wherein the computer system further includes cache storage; and

wherein the separately reserving includes separately locking respective cache lines associated with the plural locations.

51. The method of claim 50,

wherein the cache storage includes a coherently maintained set of caches including ones respectively associated with each of the plural processors.

52. The method of claim 47,

wherein the instruction implements a compound Compare-and-swap (nCAS) operation.

53. The method of claim 47,

wherein the instruction implements a Double Compare-and-swap (DCAS) operation.

54. The method of claim 47,
wherein at least one of the plural locations is identified by an operand of the instruction.

55. A method of operating a processor, the method comprising:
in response to execution by the processor of a single instruction that separately accesses plural memory locations,
accessing a first one of the memory locations; and
signaling a fault corresponding to a second one of the memory locations depending on a result of the first memory location access.

56. The method of claim 55, further comprising:
in response to the execution of the single instruction,
reserving the first and second memory locations;
detecting the fault as part of the second memory location reservation; and
signaling the fault, if at all, only upon success of the first memory location access.

57. The method of claim 55,
wherein the first memory location access includes comparing a value read from the first memory location to a corresponding test value; and
wherein the signaling is performed, if at all, only upon success of the comparing.

58. The method of claim 55, further comprising:
reserving the first and second memory locations; and
signaling the fault corresponding to the second memory location, if at all, based on a value read from the reserved first memory location.

59. A processor that implements an instruction that addresses first and second memory locations but for which signaling of a fault corresponding to the second memory location depends on a value read from the first memory location.

60. The processor of claim 59,
wherein, unless the value read compares to a test value, no fault corresponding to the second memory location is signaled.

61. The processor of claim 59,
wherein the instruction is a compound compare-and-swap instruction.

62. The processor of claim 59,
wherein the instruction separately reserves the first and second memory locations;
wherein the fault, if any, corresponding to the second memory location is detected upon reservation thereof, but signaled, if at all, based on the value read from the reserved first memory location.

63. A computer program product encoded in at least one computer readable medium, the computer program product comprising:
a set of instructions executable on a processor,
the set of instructions including at least one instance of an instruction directing the processor to access first and second memory locations but for which signaling of a fault corresponding to the second memory location depends on a value read from the first memory location.

64. The computer program product of claim 63,
wherein the instruction further directs the processor to separately reserve the first and second memory locations; and
wherein the fault, if any, corresponding to the second memory location is detected upon reservation thereof, but signaled, if at all, based on the value read from the reserved first memory location.

65. The computer program product of claim 63,
wherein the instruction further directs the processor to reserve the first and second memory locations in a predefined order in accordance with a fixed total order of memory locations.

66. The computer program product of claim 63,
wherein, unless the value read compares to a test value, no fault corresponding to the second memory location is signaled.

67. The computer program product of claim 63,
wherein the instruction is a compound compare-and-swap instruction.

68. The computer program product of claim 63,
wherein the at least one computer readable medium is selected from the set of a disk, tape or other magnetic, optical, or electronic storage medium and a network, wireline, wireless or other communications medium.

69. An apparatus comprising:
a memory store;
means for accessing in response to a single instruction, first and second locations of the memory store; and
means for signaling, if at all, a fault corresponding to the second location based on a value read from the first location.